

1194.22(1): Scripts

Scripts (cont)

- Definition

- When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology

Scripts (cont)

- Issues

- A screen reader will often read the content of the script itself in a meaningless jumble of numbers and letters. Although this jumble is text, it cannot be interpreted or used.

Scripts (cont)

- Example 1: Links

- Text link

- the following link invokes a JavaScript function called myFunction:

- ```
Start
myFunction
```

- This technique does not cause accessibility problems for assistive technology because the link is text

# Scripts (cont)

- Example 1: Links (cont)

- Using Images

- `<a href="javascript:myFunction();"></a>`
    - This coding doesn't allow the assistive technology user to understand what the link is about. The screen reader will say "Link graphic"

# Scripts (cont)

- Example 1: Links (cont)

- Using Images (cont)

- Coding Option 1: Use the “TITLE” attribute:

```
<a title="this link starts myFunction"
href="javascript:myFunction();">
```

This is not fully supported by assisted technology and not recommended. There is a violation of 1194.22(a). The image needs an ALT attribute.

# Scripts (cont)

- Example 1: Links (cont)

- Using Images (cont)

- Option 2 (Better): Use the “ALT” attribute:

```

```

Recommended because assistive technology can read the “ALT” attribute and it satisfies 1194.22(a)

# Scripts (cont)

- Example 2: Events

- onMouseOver & onMouseOut

- Issue: These event handlers can't be accessed by the mouse or keyboard. A screen reader simply bypasses them entirely.
    - Solution: Provide an alternative method of access



# Scripts (cont)

- Example 2: Events (cont)

- onMouseOver & onMouseOut (cont)

- Example: Displaying a pop-up definition

Incorrect:

```

 Definition
```

Correct:

```
<a href="definition1.htm"
 onMouseOver="show('definition1')">
 Definition
```

# Scripts (cont)

- Example 2: Events (cont)
  - onMouseOver & onMouseOut (cont)
    - The user can now use the mouse, keyboard, and the screen reader will read the link
  - onChange
    - Issue: used for triggering JavaScript functions based on a selection from within a <select> tag
    - Avoid using this event and use the onClick event handler associated with a link or button that is adjacent to a <select> tag

# Scripts (cont)

- Example 2: Events (cont)

- onChange (cont)

- Reason: An onChange event occurs whenever the object changes. The screenreader and keyboard user comes to the form control and uses the Up and Down Arrow to look through the selections. When the selection changes from the first object to the second object, an onChange event occurs. The user is never able to get pass the first option.

# Scripts (cont)

- Example 3: Menus (cont)
  - JavaScript menus are designed so that with one click, the user will be able to get the information they are searching for on the web site.
  - JavaScript menus use multiple windows that open up on the screen for navigation or multiple drop down selections.

# Scripts (cont)

- Example 3: Menus (cont)
  - JavaScript menus are primarily a mouse operation using an onMouseOver event. A user using a keyboard or screen reader will not be able to access the menu.
  - Solution:
    - On the first onMouseOver event, for each navigation area, create a hyperlink. This hyperlink will go to an alternative page with hyperlinks to the navigational areas in that menu structure.

# Scripts (cont)

- Example 3: Menus

- Solution (cont)

- Do not create a link to the site map. Create individual pages with links for each main navigational structure.
    - This will allow the user to use the keyboard or assistive technology to navigate the menu.

# Scripts (cont)

- Example 4: Pop-up Windows
  - Do not invoke a window using the JavaScript window function. Use the “window.open” function that calls a browser window.
  - Screen reader and keyboard users cannot access a non-Windows event handler because focus hasn’t been set through the Windows Application Protocol Interface (API).
  - Ensure that there is a method to close the window without using ALT-F4. Depending on the window, use a “Close button” or Windows “close” or “confirmation” buttons.

# Scripts (cont)

- Common Errors

- Using onChange instead of onClick
- Using onClick without a submit button or image
- Creating JavaScript menus without alternative navigation
- Calling window or dialog box without a method to close it.
- Calling a JavaScript window that does not set focus through the Windows API.